ELECTRICAL ENGINEERING: THE SECOND CENTURY BEGINS

Edited by
HARLOW FREITAG

THE COMPUTER EVOLUTION

Ralph E. Gomory

I am usually very pessimistic about predicting the future. Things vary a great deal in their predictability. For example, the motion of the planets is notoriously predictable. But if you pick up a stone while standing on a hill and throw it down the hillside you do not know where it will end up—the surface is too complicated, too bumpy, too rough, and the outcome too dependent on detail.

When it comes to predicting the course and impact, especially the impact, of future technologies, I think it is a problem of the second type, like the stone. I do not know how to do that predicting, and I am not sure anyone does. So I will confine myself to talking not so much about prediction but about some of the technical forces which are currently at work. The outcome of those forces acting in our society I will leave to your imagination, with confidence that you will reach a variety of different conclusions.

One of those forces is the development of the technology of smallness. I am using unfamiliar words to describe something whose present-day manifestation is, first of all, the semiconductor industry. One can say that the reason why computers make so much progress in their interior workings—and I will distinguish in this talk between their interior workings and the problems of getting things in and out of them—is basically due to the fact that they are only bit handlers. All they do is make and compare or occasionally add bits. As long as you can read it and write it, a small bit is as good as a big bit; and if you can make it out of less silicon, it is cheaper and usually faster. So there is a fundamental motivation to make things small.

Consequently, we see the evolution of ever better lithographic devices, ever cleaner areas, and the increasing problem of keeping those areas free of tiny particles which have the capability of disrupting the functioning of very small devices. Perhaps soon we will have to exclude people from these areas as people seem to emanate small particles all the time. We are creating this technology of smallness in the area of semiconductors.

But the technology of smallness is more than the familiar semiconductor example. A second aspect of making things small that is very important for computers, though perhaps less familiar, is the problem of fabricating the interconnections, or packaging. If chips are to talk to each other—and I think that will be necessary for quite a while—they have to somehow transmit their signals over wires, and those wires need to be very small.

The problem of chip interconnecting really comes from the problem of laying the interconnecting wires down between the chips, getting them up and down to the chips, and somehow building an underlying structure, the package, that is capable of supporting those wires. This in itself is an enormous task. Today, for high-speed computing, this structure, the first-level package, is 33 layers of ceramic with wiring in between. All those layers are needed to allow the degree of wiring required to interconnect the chips.

It is necessary in such a package to have very smooth surfaces. And this notion of smoothness is very closely connected with that of smallness. If you have a rather coarse wire, you can have a rather bumpy ceramic surface and you can lay the wire down on it and the bumps will make the edges look a little ragged, but it will not part the wire. On the other hand, if you want to continue making progress in miniaturization and deal with ever finer wire, you must make concurrent progress in the smoothness of that surface. And this opens up the question of how do you make supersmooth ceramic, which opens up the question of how do you make the little particles that make up the ceramic of a uniform size, and how do you make them fall together in structures that end up being very smooth?

Another aspect of miniaturization, which is not strictly semiconductor, deals with the making of disks. Again, making progress by making bits small dominates the technical problems in disks. Disks are basically like phonographs. They have an arm which swings out to some position over the disk and then tries to sense the magnetized areas of the disk as they rotate by underneath it.

The way to make progress in disks is to make the bits in the surface (the magnetized areas) smaller and smaller. This unfortunately means that the head must fly closer and closer in order to pick up their signals. And the coil structure, which is a vital part of the head, has to get smaller and smaller. Fig. 1 shows a coil structure in a head. The coils are made by the same techniques used to make semiconductors; miniaturization processes reach out beyond the area of semiconductors.

To understand how demanding the smoothness and smallness issues are in this area of computing, I might add that today's heads are passing over the surface at a height that is a fraction of a wavelength of light.

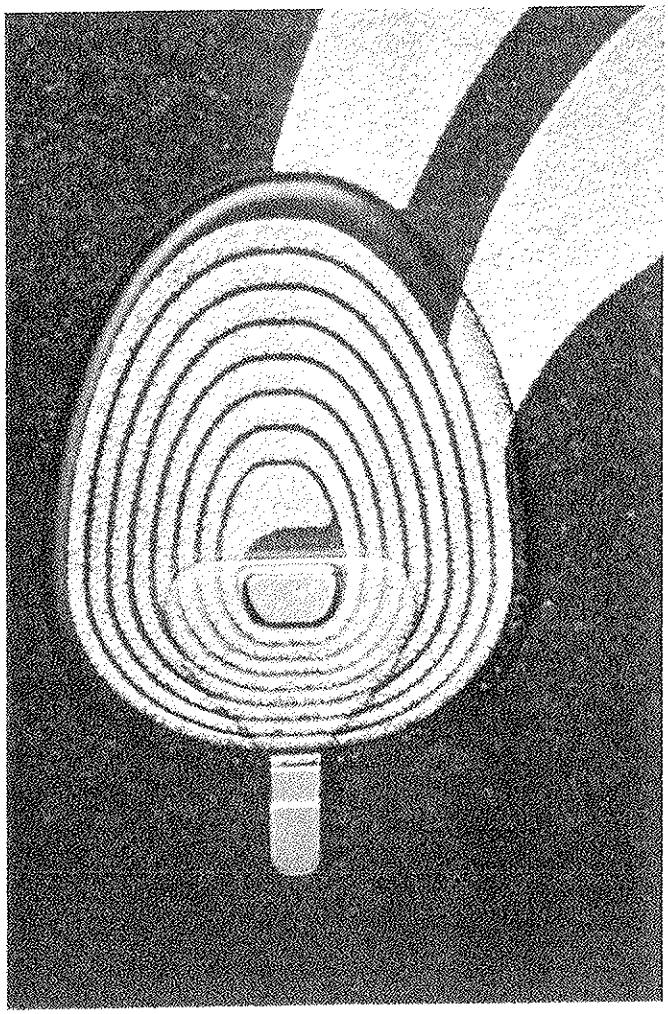

Fig. 1: Thin-film magnetic head.

Curiously enough, we deal here with dimensions smaller than those typical of semiconductors.

An analogy is useful in appreciating the small dimensions in disks. If you were to scale the head and disk up until the head was the size of a 747 aircraft, the 747 would be flying over the surface (the surface being the disk surface) at a height of only a fraction of an inch, and at full speed. Clearly, it is very important not to have any bumps.

Again, we encounter the need for smoothness in order to have smallness. And smallness is what you want when you are dealing with bits.

The technical problems of disks would be much easier if the 747 did not have to fly that close to the surface, if it could just rise up a yard or two. With magnetism, we do not see how to do that. But if we substitute a laser beam for magnetism so that the 747 can shoot little holes in the surface, or otherwise mark the surface, you ease that difficulty. Of course, you create other difficulties instead.

There are technical difficulties associated with optical storage in its various forms. For example, to write bits directly onto the surface with the laser beam, we need to bring about some sort of a change in the material, preferably reversible change, so that it is not a write-once device. One of the candidate phenomena is phase change, that is, you will crystallize the material. But in order to erase it and rewrite it, you have to uncrystallize it. So you run into a host of material problems having to do with the creation and subsequent abolition, and creation and abolition, perhaps a billion times, of very small crystalline areas.

We are seeing that the desire for information processing is creating a whole host of scientific and engineering challenges having to do with smallness and smoothness, and problems having to do with the fine structure of materials. In dealing with these problems we will have created a vast array of tools, fabrication ability, and materials understanding. This knowledge, once created, will probably not be limited in its applicability to information processing.

For example, one interesting development is the use of microfabricated structures for the separation of uranium isotopes. Basically, once you know how to make very tiny structures, it becomes possible to make very tiny metal nozzles—nozzles with a radius of curvature of 3 microns. Then, if you shoot gas around this curve, which is a very tight curve by normal standards, the separation effect for the isotopes is very intense. Indeed, you can build a whole cascaded sequence of these since they are so tiny. All together, they have very intense separation power.

Microfabrication techniques, once created, may extend to other areas. Some of the processing needed in packaging encourages you to use lasers to promote, for example, electrodeposition at a particular spot. Electroplating in the presence of a laser beam goes forward more rapidly than electroplating without it. Therefore you can create, so to speak, spots of metal by the use of a laser beam. Again, we see this notion of doing things very locally and understanding in some detail what is happening there. This notion of using certain kinds of lasers for localized effects is a natural for eye surgery, and is being explored in that realm.

I think one of the things that we are going to see is a continual pressure to learn how to microfabricate, driven by the information processing—a small bit is better than a big bit notion. The microfabrication, once created, could develop into an industry of its own, as if you had created a steel industry or something of that sort.

Within the computer, miniaturization can be regarded as a dominant means of progress. All that computing goes on inside the computer, but eventually you have to do something with it. You have to make it visible on the screen or you have to print with it. You somehow have to deal with the human scale and show things that are visible to the human eye. At this point, progress becomes harder, that is, there is no simple high road of progress like miniaturization once you reach the human scale.

However, because there is enormous progress in information processing within the machine, enormous demand is created for progress in printers and displays, not to mention more unconventional schemes for input and output.

In a variety of ways, we can confidently predict in the immediate future the improvement of displays. Certainly in 10 years we will all have high-resolution displays, 5–10 million dot displays. They may be flat, if flatness matters for other than portability reasons. For instance, you may want the display to lie down rather than to stand up stiffly in front of you. You may want to poke at the screen with its high resolution and give it a few bits of information. Color should also be a part of this picture because it is a way of conveying information.

There is an enormous challenge. With all the information being carried around inside computers, somehow you have to get it across to the user. And that gateway to the user, which is what the display or the printer is, is really worth improving. The technical capabilities for doing that are there.

The same forces drive printing, and there is a proliferation of new printing technologies. Certainly impact printing will continue. Electrophotography is a tremendous and major printing technology. But we also see, mainly driven by new demands on printing, the emergence of a whole host of other technologies such as ink jet and various forms of thermal printing. The new demands come from the personal use of computers. It does not matter if the printer prints 30,000 lines a minute, which in fact some printers do: what does matter is that it goes fast enough for one user. It does matter that it is quality printing, because the user, not some anonymous other person, reads it: and it matters that it is quiet, because it is sitting right next to the user.

These are the demands that have spawned, for example, ink jet or IBM's Quietwriter technology. And again, because you may want to print both beautiful font and high resolution for pictures, you need an all points addressable rather than a character printer.

There is progress in these areas, in some sense as a derived progress from the progress brought on by smallness. The demand is created by smallness and affects the human interface.

Let us assume that all this progress takes place. Ten years from now with these forces acting, I think we can safely predict that the ordinary mass-produced workstation or personal computer (the equivalent of today's mass-produced PC) will be a powerful engine. It will have, let us say, a single-chip microprocessor in the 10–20-mips (million instructions per second) range, which means as large as today's largest ordinary machines. It will have 16 megabytes of memory. It will have a beautiful CRT or other display of high resolution. And it will have a high-quality printer. Also, there will probably be removable optical storage of two gigabytes, and half a gigabyte of magnetic recording technology.

What I mean by a beautiful printer, which we do not take for granted today but we will within 10 years, is the kind of quality shown in Fig. 2. This is the quality of printing that can be done today, and this picture happens to have been created by an ink jet. It could be created by other means. The challenge is only to make it economically viable.

In addition to these microprocessors, there will be large systems, 100- or 150-mips systems, employing all the advanced technology, that is, there are chips with a powerful interconnect system, and all are jammed into a few cubic inches. However, this is where you start to see that there are many difficulties. Aside from the limits already described at the chip level, the business of jamming all this material into a few cubic inches, which you must do because of speed of light consideration, will be a limiting factor on the speed of very large multichip machines. I think that in the range of, for example, 150 mips, it will be very difficult to make progress beyond that, because there are engineering barriers, and very difficult ones, to get all that power in and out of the system.

Fig. 2: Ink jet printing.

Will that power suffice for the future? I think there are many applications which call for far more than that. I am going to give you a list of driving forces. They will be forces driving computers, and in most cases they will call for the creation of either special-purpose or highly parallel machines. I will use the term "highly parallel" to cover them all, because machines can be parallel in many ways. Some special-purpose machines are merely parallel machines that are parallel at a very low level.

Number one on the list are the conventional applications—all those machines that do data processing. The demand for the large machines, the mips, is growing faster than our ability to make uniprocessor (single-processor) machines for conventional applications. Already, the commercial machine is slowly evolving into a multiheaded machine.

Second, the desire to have a high degree of reliability or availability, which is most easily obtained by duplication in the processor and in the storage elements, again drives you toward multiheaded machines.

Third, there is the promise of wholly new and exciting applications, such as very large scientific calculations, which people are already doing, and for which they are building special-purpose machines. One of the machines that we are building in our research labs is to do quantum chromodynamic calculations. It is relatively easy to build a special-purpose machine, and it tends to be successful. If you name the problem, I will give you the machine. And it will go very fast. We should not, however, confuse this with the problem of devising a general-purpose parallel machine or multipurpose parallel machine, these being more difficult and requiring more speculative endeavors.

In addition to massive scientific and engineering calculations, there are, for example, calculations of design automation for which we have built special machines to simulate the logic of computers. One of the ways to reduce the design cost is to build a special-purpose machine for that very demanding purpose. And again, if you have one special purpose, you can build a machine for the purpose.

Another challenging area is artificial intelligence. One application is what is called an expert system. Fig. 3 gives a very simple example. Suppose you have a data base of facts about airlines and their flights. You may want to ask the system a variety of questions, preferably with minimal programming. One of the things that you can store is a number of facts about the system other than data, which are simple rules about the way it works.

For example, if a flight goes to the destination on the day that you want it to, then it is a possible flight for your purposes—a very trivial remark. Or if the flight is possible and if it offers a special reduced fare and a few other conditions, then it is one you might be interested

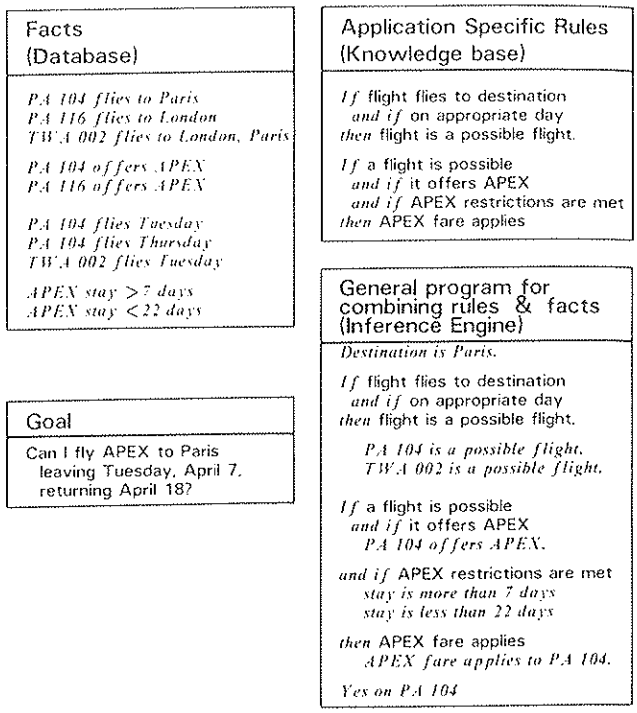

Fig. 3: Expert systems.

in. You can write down a string of rules like this. Then, if you ask questions like what flights go to Paris on the day I want to and give me an APEX fare, the system can start going through the rules. The first rule finds what flights listed in the data base are leaving on the desired day and have Paris among their list of destinations. Then the restricted list of flights is tested against your other conditions.

This kind of processing is a very trivial example of what is called a rule-based system. I want to point out a couple of things about it. It opens up the possibility of asking many different questions about a set of data provided that you have built in the proper set of rules.

Second, it is very computation-demanding. This trivial example does not show it but you will normally find that these systems end up doing a great deal of relatively blind searches. In some sense, this is another extension of the way we make progress in programming; that is, we do less and less art and use more and more computing power. However, given the progress of machines, this is fundamentally the right way to go. We have an expert system in one of our labs that parses English and takes about 20 million instructions per sentence; it is a very demanding user of machines.

Fig. 4 shows again the example of Fig. 3 only written in PROLOG, which is one of the languages used for the expert systems. You can, as a first approximation, think of the fifth-generation effort in Japan as the development of parallel and special-purpose machines and the software to do this kind of work. There are a lot of other things on it, like natural language, but I would say this is the technical core of that effort.

program to book a flight

```
book-flight<-read(Origin,Destination,Dept-date,Ret-date)
    &flight-list(flight(Origin,Destination,*,*,*,*,*),List)
    &check-flights(List,Possible-flights,Dept-date,Ret-date)
    &sort-by-fare(Possible-flights,Flights)
    &show(flights)
    &read(f-number)&book(f-number)
```

rules for fares and restrictions

```
check-flights(Flight,Rest,Possible-flight,Other flights,Dept-date,Ret-date) <-
    fares(Org,Dest,Airline,Fare-class,Fare,Restriction) &
    check-class(Fare-class,Restriction,Org,Dest,Dept-date,Ret-date) &
    check-flights(Rest,Other-flights,Dept-date,Ret-date)
check-class(y,ex/Note-nbr,Org,Dest,Dept-date,Ret-date) <-
    restrictions(Note-nbr,Start-date,End-date,Min-stay,Max-stay) &
    length-of-stay(Dept-date,Ret-date,Los) & ge(Los,Min-stay) & le(Los,Max-stay)
```

database of flights, fares and restrictions

```
flight(newyork,paris,1300,jfk,2245,cdg,af,002,r).
flight(newyork,paris,1845,jfk,0745,ory,pa,114,y).
fares(newyork,paris,af,r,2060.00,fixed).
fares(newyork,paris,pa,y,629.00,ex/4157).
restrictions(4157,6/19,6/30,14D,2M).
restrictions(4409,5/15,9/14,7D,3M).
```

Fig. 4: PROLOG example.

Another tremendous user of mips and driver for mips that probably cannot be supplied by the ordinary machine is the user interface. There are several: one is a sort of unconventional input/ output, for example, speech recognition. It is a tremendous mips burner. The best estimates I can make on the subject are that to do any kind of reasonable continuous speech, many hundreds of mips would be required. These estimates are suspect because to do continuous speech is an ill-defined term. One has to be concerned with what error rate, over what vocabulary, in perhaps a limited range of discourse, or other concerns. The main message is that trying to recognize speech is another enormous mips demander. Similarly, the understanding of hand-written input and of natural language is also, I believe, feasible, and both are enormous mips burners.

Finally, there is the direct manipulation of objects in place of programming, which I will mention very briefly. I am going to put it in the somewhat exaggerated way that programmers are really used to dealing with a conceptual structure which is the machine. For example, they talk about memory locations. They imagine they are putting things in the memory registers. Now, there are new departures and new models that allow you, instead of the machine, to manipulate the objects.

For example, take the desk top image in which you do things by pushing around on a desk top things that look relatively real, such as putting a piece of paper in a wastebasket. Actually, what you do is a mixture of symbolic processing and moving images. This is a significant input/output operation that will again be very demanding in mips because there is so much to be done that requires enormous power to translate it into usable form. That transformation, whether it is through speech recognition or maneuvering objects, is another tremendous demander of mips.

All of this leads to the study of unconventional architectures and, especially, parallel architectures, because it is basically out of parallelism that you get the increase in mips for your special purpose. It is nontrivial to try to develop the proper parallel machines.

It is relatively straightforward to develop some types of special-purpose parallel machines. But the notion of a general-purpose parallel machine is a much more elusive and difficult thing. There will be tremendous needs and pressures to create an understanding of parallel algorithms and to create software to run these parallel machines.

Interconnecting multiple discrete processors will emerge as a discipline of its own. It is not simply a question of hanging processors on a bus. The interconnect structure will probably be critical and will have a great deal of structure itself. It will be more than a simple switch in all probability. The need for mips will not be satisfied by the conventional machine. Those pressures will create new architectures, and a whole new family of progress is needed.

In this direction, some of the most demanding users of this new power will be software itself. The creation of software is one of the most complex things that people do. It is a somewhat maligned subject because of its proximity to hardware. Hardware makes progress at an extraordinary rate because of the smallness issue. We can make progress in computers through making things small. Most technologies do not have such a magic formula. Automobiles do not and almost anything else you think of does not. Software is in the ordinary category. Its misfortune is that it is sitting next to something very unusual. I think we should stop demanding of software that it make the extraordinary progress of its neighbor. It is ordinary. It is its neighbor that is extraordinary. Software benefits from that hardware progress, and it will continue to do so. It will continue to develop as an engineering discipline of its own. Better interfaces and an increase in computing power will benefit it, and it will develop its own engineering procedures. The software generation procedures will become both more disciplined and more perfect.

Finally, I would like to say a few words about robots. Those of us who were brought up on H. G. Wells remember the clanking mechanical monsters that did such wonderful things or such evil things, depending on what story it was. As a child, I wondered why it was we couldn't build those when I could see all around us equally complex machines. After a lot of thought, I reached the conclusion that something was lacking. I

mean that if I imagined myself building it, there was no way to direct it.

When I became the director of research for IBM in 1970, we had labs full of people working on intelligence, but none on mechanical motion. After a while, we started a robot project, and today we have a small robot business.

The essential ingredients are there. The ability to power mechanical motion has been there since the industrial revolution. The ability to do a great deal of thought-like work is here today. It is relatively straightforward—everything is straightforward in the perspective of 100 years—to equip these creatures with sensors and with vision. Robots and mechanized production are definitely coming.

One of the forces at work is the creation of a technology of miniaturization. It will continue to enable us to make progress for some time, and it will have its own consequences. We will, by the extrapolation of the doctrine of smallness, have machines of enormous power and interfaces providing great ease of use. Nevertheless, as we approach the limit of these machines, parallelism and special machines in all their forms will become necessary and challenge us in many ways—scientifically, algorithmically, and in other ways.

Robots are a real possibility because the ingredients, physical and mental, are present. This combination of forces will have a most profound effect on the world.

Ralph E. Gomory, *IBM Fellow and senior vice president and director of research, heads IBM's Research Division, which includes laboratories in Yorktown Heights, NY; San Jose, CA; and Zurich, Switzerland. He received his Ph.D. in mathematics from Princeton after studies at Williams and Cambridge. Dr. Gomory is a member of the National Academies of Engineering and Sciences, the American Philosophical Society, and the White House Science Council. His honors include the Harry Goode Memorial Award of the American Federation of Information Processing Societies, the Industrial Research Institute Medal, the Lanchester Prize of the Operations Research Society, and the John von Neumann Theory Prize, given jointly by the Operations Research Society and The Institute of Management Science.*